

On the convergence of numerical solutions to the continuous-time constrained LQR problem

Gabriele Pannocchia¹ **David Q. Mayne**² **James B. Rawlings**³

1. Department of Chemical Engineering, University of Pisa, Italy (g. pannocchia@diccism.unipi.it)
2. Department of Electrical and Electronic Engineering, Imperial College London, UK (d. mayne@imperial.ac.uk)
3. Chemical and Biological Engineering, University of Wisconsin, Madison (WI), USA (rawlings@eng.wisc.edu)



21st International Symposium on Mathematical Programming
Berlin (Germany), August 19 – 24, 2012

Outline

- 1 Problem statement
 - Introduction and previously proposed method
 - Objectives
- 2 Theoretical foundations
 - Exploiting matrix exponentiations
 - Optimality functions
- 3 Proposed algorithm and convergence results
 - Lower bound on discretized problems
 - Conceptual algorithm
 - Implementation aspects
 - Convergence analysis
- 4 Example results
- 5 Conclusions and future work

Problem statement

The optimal control problem (OCP) $\mathbb{P}_T(x)$

- Solve repeatedly for different $x \in \mathbb{R}^n$

$$\min_{u(\cdot)} V_T(x, u(\cdot)) \triangleq \int_0^T \ell(x(t), u(t)) dt + V_f(x(T)) \quad \text{s. t.}$$

$$\dot{x} = f(x, u) \triangleq Ax + Bu, \quad x(0) = x$$

and the **control constraint**: $u(t) \in \mathbb{U}$

- Quadratic** cost functions: $\ell(x, u) \triangleq \frac{1}{2}(x'Qx + u'Ru), \quad V_f(x) \triangleq \frac{1}{2}x'Px$

Main assumptions

- R and P **symmetric positive definite** (SPD), Q SPSD, and

$$\mathbb{U} \triangleq \prod_{i=1}^m \mathbb{U}_i \quad \text{where} \quad \mathbb{U}_i \triangleq [u_{\min}^i, u_{\max}^i]$$

- (A, B) **stabilizable**, (A, Q) **detectable**, T **large enough** that $x^0(T) \in X_f$, invariant set associated with $V_f(\cdot)$: **terminal constraint** $x(T) \in X_f$ is **omitted**

Input parameterizations and unconstrained analysis

Three possible holds in $t \in [t_j, t_{j+1}]$

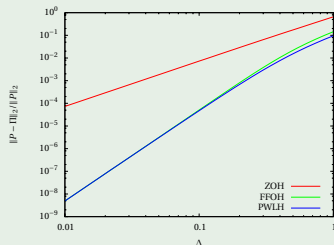
- **ZOH**: $u(t) \triangleq u_j$
- **PWLH**: $u(t) \triangleq u_j + s_j(t - t_j)$ or let $s_j \triangleq \frac{v_j - u_j}{t_{j+1} - t_j}$:

$$u(t) = u_j(1 - r(t)) + v_j r(t) \quad \text{where} \quad r(t) = \frac{t - t_j}{t_{j+1} - t_j}$$

- **FFOH**: like PWLH but **continuous** at each t_j , i.e. $u_j = v_{j-1}$

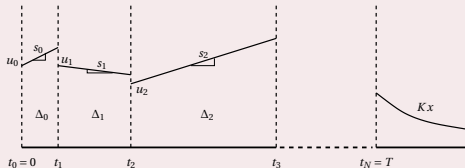
Convergence of the optimal cost without control constraint

- **True optimal** cost is well-known:
 $V_T^0(x) = \frac{1}{2} x' P x$, P solution to ARE
- **Optimal** cost under each **hold** i :
 $V_T^i(x) = \frac{1}{2} x' \Pi_i x$, Π_i solution to a DARE
- $\Pi_i \rightarrow P$ as $\Delta \triangleq t_{j+1} - t_j \rightarrow 0$ in **4th** order for **PWLH and FFOH** and 2nd order for ZOH



Previously proposed method¹

Solution of the optimal control problem



- Use N **uneven** intervals with a chosen hold (ZOH, PWLH or FFOH)
- Solve **exactly** the OCP as a **QP**
- Bisect **all intervals**, until the cost “stops” decreasing

Main features

- All QP terms are **precomputed offline** for gradually refined grids:

$$\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}' \mathbf{H} \mathbf{w} + \mathbf{w}' \mathbf{Q} \mathbf{x}, \text{ s.t. } \mathbf{A} \mathbf{w} \leq \mathbf{b}$$

- **Fast online** implementation

¹G. Pannocchia, J.B. Rawlings, D.Q. Mayne and W. Marquardt “On Computing Solutions to the Continuous Time Constrained Linear Quadratic Regulator”, IEEE Trans. Auto. Contr., 55 (9), 2192–2198, 2010

Objectives of this work

Main goals

- Devise an **adaptive grid** strategy
- Prove **convergence** towards the optimal solution (with control constraints)
- Provide **degree of suboptimality** for finite iterations

Working tools

- **Matrix exponentiation** formulas to **avoid ODE** integration
- **Optimality functions** for discrete-time and continuous-time CLQR problems
- **Functional** analysis

Our focus

- The algorithm is intended as a **replacement** for **discrete-time MPC**
- We solve $\mathbb{P}_T(x)$ many times, i.e. for any given **current state** $x \in \mathbb{R}^n$ that occurs in closed-loop operation

Who needs ODE solvers anyway? ... ZOH case²

Consider a generic interval $[0, \Delta]$

- It is **well-known** that:

$$x(\Delta) = A_{\Delta}x(0) + B_{\Delta}u(0) \quad \text{where} \quad A_{\Delta} = e^{A\Delta}, \text{ and } B_{\Delta} = \int_0^{\Delta} e^{A\tau} B d\tau$$

$$\int_0^{\Delta} (x(\tau)' Q x(\tau) + u(\tau)' R u(\tau)) d\tau = x(0)' Q_{\Delta} x(0) + 2x(0)' M_{\Delta} u(0) + u(0)' R_{\Delta} u(0)$$

- How to compute $\int_0^{\Delta} e^{A\tau} B d\tau$ and $(Q_{\Delta}, M_{\Delta}, R_{\Delta})$ **without ODE solvers**?

All at once... (faster and much more accurate)

Define C and its **exponential**: $C \triangleq \begin{bmatrix} -A' & I & 0 & 0 \\ & -A' & Q & 0 \\ & & A & B \\ & & & 0 \end{bmatrix}$, $e^{Ct} \triangleq \begin{bmatrix} F_1(t) & G_1(t) & H_1(t) & K_1(t) \\ & F_2(t) & G_2(t) & H_2(t) \\ & & F_3(t) & G_3(t) \\ & & & F_4(t) \end{bmatrix}$ then:

$$e^{A\Delta} = F_3(\Delta) \quad B_{\Delta} = G_3(\Delta)$$

$$Q_{\Delta} = F_3'(\Delta) G_2(\Delta) \quad M_{\Delta} = F_3'(\Delta) H_2(\Delta) \quad R_{\Delta} = [B' F_3'(\Delta) K_1(\Delta)] + [*]'$$

²C.F. Van Loan "Computing Integrals involving the Matrix Exponential", IEEE Trans. Auto. Contr., 23 (3), 395–404, 1978

Who needs ODE solvers anyway? ... PWLH case

Consider a generic interval $[0, \Delta]$

- Let $w(0) = (u(0), v(0))$. Assume PWLH:

$$u(t) = u_j(1 - r(t)) + v_j r(t) \quad \text{with} \quad r(t) = \frac{t}{\Delta}$$

- Obtain** without ODE solvers the discretized **system** and **cost matrices**:

$$x(\Delta) = A_{\Delta} x(0) + B_{\Delta} w(0)$$

$$\int_0^{\Delta} (x(\tau)' Q x(\tau) + u(\tau)' R u(\tau)) d\tau = x(0)' Q_{\Delta} x(0) + 2x(0)' M_{\Delta} w(0) + w(0)' R_{\Delta} w(0)$$

Define a suitably augmented system... and we're done

- Let: $A^* \triangleq \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}$, $B^* \triangleq \begin{bmatrix} B & 0 \\ -\frac{1}{\Delta} & \frac{1}{\Delta} \end{bmatrix}$, $Q^* \triangleq \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$.
- Define C and $\exp(C\Delta)$ **as in ZOH** and obtain:

$$A_{\Delta} = F_3(\Delta)_{(1:n,1:n)} \quad B_{\Delta} = G_3(\Delta)_{(1:n,:)} \quad Q_{\Delta} = (F_3'(\Delta) G_2(\Delta))_{(1:n,1:n)}$$

$$M_{\Delta} = (F_3'(\Delta) H_2(\Delta))_{(1:n,:)} \quad R_{\Delta} = \begin{bmatrix} \frac{1}{3} R \Delta & \frac{1}{6} R \Delta \\ \frac{1}{6} R \Delta & \frac{1}{3} R \Delta \end{bmatrix} + [B' F_3'(\Delta) K_1(\Delta)] + [*]'$$

Generalities on optimality functions

Preliminaries: the space of control and state trajectories

- The control $u(\cdot)$ is assumed to lie in \mathcal{U} defined as:

$$\mathcal{U} \triangleq \{u : [0, T] \rightarrow \mathbb{R}^m \mid u(\cdot) \text{ **measurable** and } u(t) \in \mathbb{U} \text{ for all } t \in [0, T]\}$$

- For any $1 \leq p \leq \infty$, we observe that $\mathcal{U} \subseteq L_p$, Banach space defined as:

$$L_p \triangleq \{u : [0, T] \rightarrow \mathbb{R}^m \mid u(\cdot) \text{ **measurable** and } \|u(\cdot)\|_p < \infty\}$$

- $u(\cdot) \in \mathcal{U}$ implies that $x(t) \triangleq \phi(t; x, u(\cdot))$ is **absolutely continuous**

Seeking an optimality function

- Given the initial state $x \in \mathbb{R}^n$ and a control $u(\cdot) \in \mathcal{U} \subset L_p$, we seek a **continuous, nonpositive** function $\theta : \mathbb{R}^n \times \mathcal{U} \rightarrow \mathbb{R}_{\leq 0}$ such that
 - ▶ $\theta(x, u(\cdot)) < 0$ if $u(\cdot)$ is **not optimal** for $\mathbb{P}_T(x)$
 - ▶ $\theta(x, u(\cdot)) = 0$ if $u(\cdot)$ is **optimal** for $\mathbb{P}_T(x)$ – “optimal” rather than “locally optimal” because $\mathbb{P}_T(x)$ is **strictly convex**

An optimality function for $\mathbb{P}_T(x)$

Cost gradient w.r.t. $u(\cdot)$ and its relation to the Hamiltonian

- Given $(x, u(\cdot))$, let $x(t) = \phi(t; x, u(\cdot))$. Define $\lambda : [0, T] \rightarrow \mathbb{R}^n$, solution to the **adjoint equation**:

$$-\dot{\lambda}(t) = A' \lambda(t) + Qx(t) \quad \lambda(T) = Px(T)$$

- Define the **Fréchet derivative** of $V_T(\cdot)$ w.r.t. $u(\cdot)$ as $g(x, u(\cdot)) = D_u V_T(x, u(\cdot))$
- For any $t \in [0, T]$, there holds: $g(x, u(\cdot))(t) = \nabla_u H(x(t), u(t), \lambda(t))$, where $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$ is the **Hamiltonian**:

$$H(x, u, \lambda) \triangleq \ell(x, u) + \lambda'(Ax + Bu)$$

Definition

- Define (since x is **fixed in the algorithm execution** we omit it) $\theta : \mathcal{U} \rightarrow \mathbb{R}_{\leq 0}$

$$\theta(u(\cdot)) \triangleq \int_0^T \langle g(x, u(\cdot))(t), u^*(t) - u(t) \rangle dt \quad \text{where}$$
$$u^*(t) \triangleq \arg \min_{v \in \mathbb{U}} \langle g(x, u(\cdot))(t), v \rangle$$

An optimality function for $\mathbb{P}_T(x)$: results and computation

Main results

- Result 1: $\theta(u(\cdot))$ is an **optimality function** for $\mathbb{P}_T(x)$
- Result 2: Since $\mathbb{P}_T(x)$ is a **convex problem**, there holds

$$V_T^0(x) \geq V_T(x, u(\cdot)) + \theta(u(\cdot)) \quad \text{where} \quad V_T^0(x) \triangleq \min_{u(\cdot) \in \mathcal{U}} V_T(x, u(\cdot))$$

Computation (... ODE solver based)

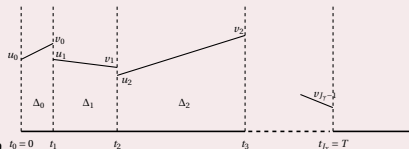
- **Integrate** (backward) the **adjoint equation** to define $g(x, u(\cdot))(t)$
- Evaluate $u^*(t)$ and **compute the integral** defining $\theta(u(\cdot))$

Note: the above steps can be computed with an **ODE** (in $n + 1$ variables) solver call

Discretized problem and its solution

Discretization (PWLH case)

- A **discretization** γ is a sequence of J_γ intervals with **disjoint interiors**:
 $I_j \triangleq \{[t_j, t_{j+1}] \mid j \in \mathbb{I}_{0:J_\gamma-1}\}$ such that
 $[0, T] = \bigcup_{j=0}^{J_\gamma-1} I_j$
- $\mathcal{U}^\gamma \triangleq \{u(\cdot) \in \mathcal{U} \mid u(\cdot) \text{ is **linear** in each } I_j\}$



Solution to the discretized problem

$\mathbb{P}_T^\gamma(x)$: $\min_{u(\cdot) \in \mathcal{U}^\gamma} V_T(x, u(\cdot))$ is **equivalent to**:

$$\min_{\mathbf{w}} V_T^\gamma(x, \mathbf{w}) \triangleq \sum_{j=0}^{J_\gamma-1} L_j(x_j, w_j) + V_f(x_{J_\gamma}) \quad \text{s.t. } x_{j+1} = A_j x_j + B_j w_j, \quad w_j = (u_j, v_j) \in \mathbb{U}^2$$

where: $x_0 = x$, $\mathbf{w} \triangleq (w_0, w_1, \dots, w_{J_\gamma-1})$, $L_j(x, w) = \frac{1}{2}(x' Q_j x + 2x' M_j w + w' R_j w)$ and $(A_j, B_j, Q_j, M_j, R_j)$ are the **discretized matrices** for interval size Δ_j

A fast and useful (discrete-time) lower bound

Optimality function for $\mathbb{P}_T^\gamma(x)$

- Discrete-time **adjoint system**:

$$\lambda_j = A_j' \lambda_{j+1} + M_j' w_j + Q_j x_j \quad \lambda_{J_\gamma} = P x_{J_\gamma}$$

and **Hamiltonian**: $H_j(x, w, \lambda) \triangleq L_j(x, w) + \lambda'(A_j x + B_j w)$

- Gradient** of optimal cost: $g^\gamma(x, \mathbf{w}) \triangleq D_{\mathbf{w}} V_T^\gamma(x, \mathbf{w}) = \{g_0, g_1, \dots, g_{J_\gamma-1}\}$ where

$$g_j = \nabla_{w_j} H_j(x_j, w_j, \lambda_{j+1}) = M_j x_j + R_j w_j + B_j' \lambda_{j+1}$$

- Optimality** function: $\theta^\gamma(u(\cdot)) = \sum_{j=0}^{J_\gamma-1} \theta_j^\gamma$ where $\theta_j^\gamma \triangleq \langle g_j, w_j^* - w_j \rangle$ and

$$V_T^{0,\gamma}(x) \geq V_T^\gamma(x, \mathbf{w}) + \theta^\gamma(u(\cdot))$$

Let Δ be the “smallest possible” interval size

- The **finest discretization** γ^Δ is that in which **all intervals** have size **equal** to Δ
- $\theta^\Delta(u(\cdot)) \triangleq \theta^{\gamma^\Delta}(u(\cdot))$ is an **optimality function** for $\mathbb{P}_T^{\gamma^\Delta}(x)$ at **finest discretization**

Overall (conceptual) algorithm

Setup

- Γ_Δ is the set of **all discretizations** in which each Δ_j is an **even multiple** of Δ
- $\gamma' \in \Gamma_\Delta$ is a **refinement** of $\gamma \in \Gamma_\Delta$ if γ' is derived from γ **bisecting some** intervals

Master algorithm

Data: $x \in \mathbb{R}^n$, $\Delta > 0$, $\epsilon > 0$, $c \in (0, 1)$, $\gamma \in \Gamma_\Delta$

Step 1: Solve \mathbb{P}_T^γ and **obtain control** $u(\cdot) \in \mathcal{U}^\gamma$. **Compute** $\theta^\Delta(u(\cdot))$

Step 2: Refine γ (repeatedly) until $\theta^\gamma(u(\cdot)) \leq c\theta^\Delta(u(\cdot))$

Step 3: If $\theta^\Delta(u(\cdot)) < -\epsilon$, **go to Step 1**. Else **go to Step 4**

Step 4: Replace $\epsilon \leftarrow \frac{\epsilon}{2}$, $\Delta \leftarrow \frac{\Delta}{2}$. **Bisect largest** interval in γ and **go to Step 1**

Comments

- At the **end of Step 1**: $\theta^\gamma(u(\cdot)) = 0$. In Step 2 γ is **refined**, thus $\theta^\gamma(u(\cdot)) < 0$
- In Step 4, we **can use** $\epsilon \leftarrow c_1\epsilon$ and $\Delta \leftarrow c_2\Delta$ where $c_1, c_2 \in (0, 1)$

Implementation aspects

Adaptive bisection strategy

- Given γ and $u(\cdot)$, if we **bisect an interval** I_j obtaining (I_{j_1}, I_{j_2}) , the **PWLH parameters** in each subinterval are: $w_{j_1} = (u_j, \frac{u_j + v_j}{2})$, $w_{j_2} = (\frac{u_j + v_j}{2}, v_j)$
- We **can easily compute** $\theta_j^\gamma = \langle g_{j_1}, w_{j_1}^* - w_{j_1} \rangle + \langle g_{j_1}, w_{j_1}^* - w_{j_1} \rangle$
- If $\sum \theta_j^\gamma > c\theta^\Delta(u(\cdot))$, we **bisect all intervals** and repeat the procedure. Else, we bisect the **smallest number** of I_j such that $\theta^\gamma(u(\cdot)) \leq c\theta^\Delta(u(\cdot))$

Stopping criteria

- As written, the Master algorithm **does not terminate**
- Possible stopping criteria can be (for some small $\rho > 0$):
 - In **Step 4**, compute the **CT optimality function** $\theta(u(\cdot))$ and stop if $\theta(u(\cdot)) \geq -\rho$
 - After **Step 1**, stop if $\theta^\Delta(\cdot) \geq -\rho$

Offline computations (performed for different interval sizes)

All **matrices** required for $\mathbb{P}_T^\gamma(x)$ and $\theta^\gamma(u(\cdot))$ are computed and stored **offline**

Convergence analysis: preliminary definitions and results

Definitions

- Let $u_i(\cdot)$ be the **control function** computed at **iteration i** of the Algorithm
- **Same meaning** for ϵ_i , γ_i and Δ_i
- Let δ_i be the size of the **largest interval** at iteration i

Preliminary considerations

- The “loop” **Step 1 to Step 3** is always executed a **finite number** of iterations
- Let \mathcal{I} index the **subsequence of iterations** in which **Step 4 is executed**
- Clearly: $(\epsilon_i, \Delta_i) \rightarrow 0$ as $i \xrightarrow{\mathcal{I}} \infty$. **Hence**, $\delta_i \rightarrow 0$ as $i \xrightarrow{\mathcal{I}} \infty$

Theorem (Continuity of the optimal solution to $\mathbb{P}_T(x)$)

$u^0(\cdot) : [0, T] \rightarrow \mathbb{U}$ is Lipschitz continuous

Convergence analysis: main results

Theorem (Convergence of the cost $V_T(\cdot)$ computed by the algorithm)

It follows that $V_T(x, u_i(\cdot)) \xrightarrow{\mathcal{J}} V_T^0(\cdot)$ as $i \rightarrow \infty$

Proof ingredients

- Let $u_i^*(\cdot)$ be the **sample-hold** version (in **PWLH** sense) of $u^0(\cdot)$ according to **discretization** γ_i
- Use **Lipschitz** continuity of $u^0(\cdot)$ to show that $u_i^*(\cdot) \xrightarrow{\mathcal{J}} u^0(\cdot)$ as $i \rightarrow \infty$, because $\delta_i \rightarrow 0$
- Since $V_T(\cdot)$ is **continuous**, it follows that $V_T(x, u_i^*(\cdot)) \xrightarrow{\mathcal{J}} V_T^0(\cdot)$ as $i \rightarrow \infty$
- Since $u_i^*(\cdot) \in \mathcal{U}^{\gamma_i}$, i.e. **feasible** for $\mathbb{P}_T^{\gamma_i}(\cdot)$, there holds

$$V_T^0(x) \leq V_T(x, u_i(\cdot)) \leq V_T(x, u_i^*(\cdot))$$

Corollary (Convergence of the control function $u_i(\cdot)$)

It follows that $u_i(\cdot) \xrightarrow{\mathcal{J}} u^0(\cdot)$ in L_p as $i \rightarrow \infty$

Illustrative example: system and algorithm parameters

System, LQ penalties and input constraints

- **Stable** system with one **slow over-damped** mode (time constant of 10) and **two fast oscillating** modes (time constant of 1)

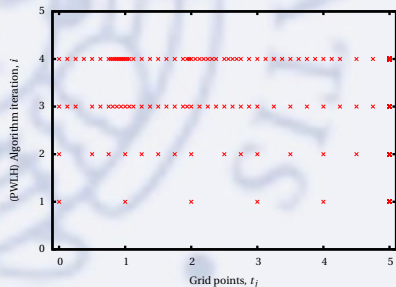
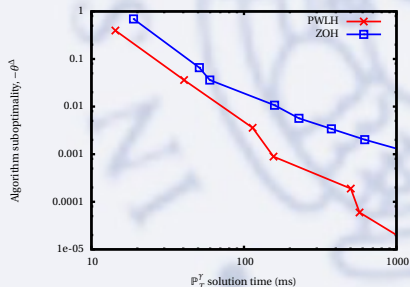
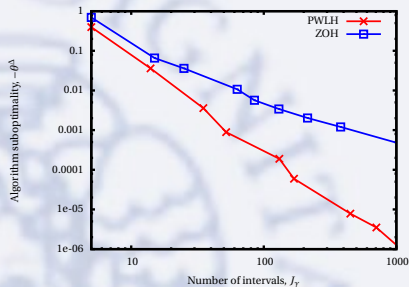
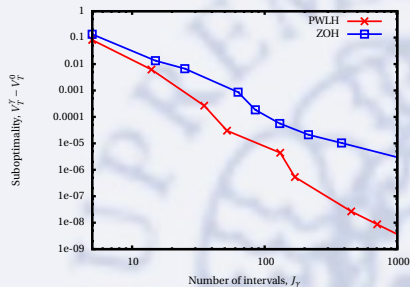
$$\frac{0.02s^2 + 5.04s + 1}{0.4s^3 + 0.84s^2 + 10.08s + 1}$$

- LQ penalties: $Q = I, R = 0.1$
- Input constraints: $u \in \mathbb{U} = [-1, 1]$

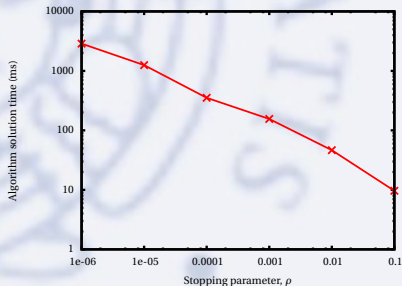
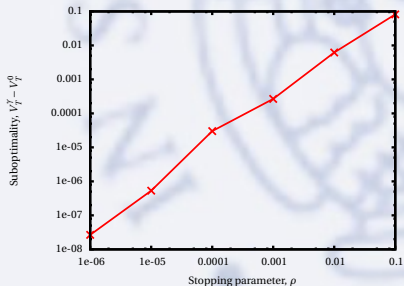
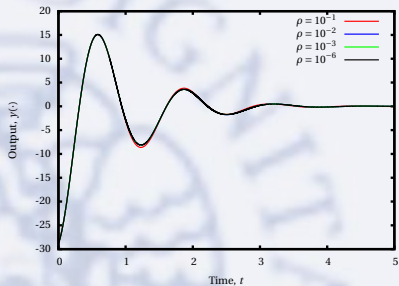
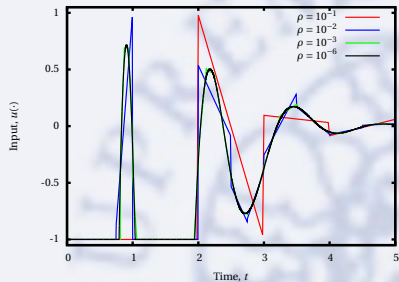
Algorithm parameters

- **Final time** $T = 5$
- Initial **discretization points** $t_j \in \{0, 1, \dots, 4\}$
- Initial **finest discretization** size $\Delta = 0.0625$ (80 intervals)
- Initial **Step 3** tolerance $\epsilon = 0.1$
- **Refinement** parameter $c = 0.8$

Illustrative example: using the conceptual algorithm



Illustrative example: using the practical algorithm (PWLH)



Conclusions and work in progress

Concluding remarks

- Proposed/ revised an algorithm for solutions to **continuous-time constrained LQR** problem
- **Adaptive** discretization and **piece-wise linear** input parameterization (constraint satisfaction and faster convergence)
- No **need for ODE solvers** in all steps (offline and online) due to **clever exponentiation** formulas
- **Optimality functions** are proposed and computed, which provide useful information for **grid refinement** and **practical stopping** conditions
- **Convergence** towards the optimal solution is **proved**

Current work

- **Fast** implementation (so far plain Octave was used...)
- Efficient (almost) **analytical** computation of **CT optimality function**
- Closed-loop **stability** and **nominal robustness**